**Online Grocery Management Capstone – Learner Implementation Guide**

**Sprint 1: ERD Design & Normalization**

**Tasks:**

1. Identify at least **8 logical entities** (Customer/User, Seller, Product, Category, Order, OrderItem, Payment, Review, InventoryLog).
2. Define attributes with PKs and simple candidate keys.
3. Design relationships with cardinalities:
   * 1:M (e.g., Seller → Products)
   * M:N via bridge (Orders ↔ Products through OrderItem)
4. Normalize to **3NF**:
   * 1NF: Atomic fields
   * 2NF: Composite PK dependencies resolved
   * 3NF: No transitive dependencies; lookup tables for categories/roles
5. Create an ER diagram including PKs, FKs, and relationships.
6. Write a justification document explaining normalization and relationship choices, plus how roles are simulated via role\_id.

**Sprint 2: SQL Schema and Reporting View**

**Tasks:**

1. Write DDL scripts in Oracle Live SQL to create all tables:
   * Roles, Users, Categories, Products, Orders, Order\_Items, Payments, Reviews, Inventory\_Log.
   * Include datatypes, PK/FK constraints, CHECK, NOT NULL, and UNIQUE.
2. Create SEQUENCEs for auto-generated IDs (ex: seq\_user, seq\_order, etc.).
3. Load sample data for Roles and Users to demo each role type.
4. Define a **reporting view** (e.g., vw\_order\_summary) that:
   * Joins Orders, Users, Order\_Items
   * Calculates total order value (SUM(qty \* price\_at\_order))
   * Groups by order\_id, username, order\_date, and status.

**Deliverables:**

* SQL script with tables, constraints, sequences, and sample data.
* A screenshot or output of the reporting view showing aggregated order totals.

**Sprint 3: PL/SQL Logic, Triggers, and Packaging**

**Tasks:**

1. **Procedures with Role Checks**:
   * add\_product: Only Sellers can insert, default approved = 'N'
   * approve\_product: Only Admins can update approved flag
   * place\_order: Only Customers can order; validates stock; inserts into Orders & Order\_Items; updates stock
2. **Trigger**: After stock\_qty update on Products, insert a record into Inventory\_Log.
3. **Exception Handling**: Use custom exceptions for:
   * Unauthorized role actions
   * Insufficient stock during place\_order
4. **Package**: Group procedures and exceptions into Grocery\_PKG.

**Demo Steps:**

* Run each procedure using user IDs for different roles, demonstrating success and error messages.
* Show trigger logging working correctly upon stock updates.

**Deliverables:**

* PL/SQL script with procedures, trigger, and package
* Test script with sample calls and outputs validating each case
* Explanation table mapping each procedure/trigger back to real-world action

**Final Submission Checklist**

| **Sprint** | **Delivered Item** |
| --- | --- |
| 1 | ER Diagram (3NF) + normalization justification + relationship mapping |
| 2 | SQL DDL + sample data + sequences + reporting view |
| 3 | PL/SQL: Procedures, trigger, package + test script + role-based behavior proof |
| Overall | Explanation of how each component (table, code, view) ties to a business scenario |